strcpy_s() and strcat_s()

Daniel Plakosh, Software Engineering Institute [vita[1]]

2005-09-27

The `strcpy_s()` and `strcat_s()` functions are defined in ISO/IEC TR 24731 as a close replacement for `strcpy()` and `strcat()`. These functions have an additional argument that specifies the maximum size of the destination and also include a return value that indicates whether the operation was successful.

Development Context

Copying and concatenating character strings

Technology Context

C, UNIX, Win32

Attacks

Attacker executes arbitrary code on machine with permissions of compromised process or changes the behavior of the program.

Risk

The `strcpy()` and `strcat()` functions are a source of buffer overflow vulnerabilities.

Description

The `strcpy_s()` and `strcat_s()` functions are defined in ISO/IEC WDTR 24731 as a close replacement for `strcpy()` and `strcat()`. These functions have an additional argument that specifies the maximum size of the destination and also include a return value that indicates whether the operation was successful.

The `strcpy_s()` function is similar to `strcpy()` if a constraint violation does not occur. In this case, the `strcpy_s()` function copies characters from the source string to the destination character array up to and including the terminating null character and then returns zero to indicate success.

The `strcpy_s()` function only succeeds when the source string can be fully copied to the destination without overflowing the destination buffer. If either the source or destination pointers are null or if the maximum length of the destination buffer is equal to zero, greater than RSIZE_MAX,[14] or less than or equal to the length of the source string, then a constraint violation occurs and the operation returns a non-zero value. Additionally, the `strcpy_s()` function will result in a constraint violation if the memory regions of the objects overlap. If a constraint violation occurs, a zero is stored in the first

1. daisy:268 (Plakosh, Daniel)

character of the destination if the destination pointer is not equal to null and the size of the destination buffer is greater than zero and less than or equal to RSIZE_MAX.

The `strcat_s()` function appends the characters of the source string, up to and including the null character, to the end of the destination string. The initial character from the source string overwrites the null character at the end of the destination string.

The `strcat_s()` function returns zero on success. A constraint violation will occur and the operation will return a non-zero value if

- either (*a*) the source or destination pointer is null or the maximum length of the destination buffer is equal to zero or greater than RSIZE_MAX or (*b*) the destination string is already full or there is not enough room to fully append the source string
- the memory regions of the objects overlap

If a constraint violation occurs, a zero is stored in the first character of the destination if the destination pointer is not equal to null and the size of the destination buffer is greater than zero and less than or equal to RSIZE_MAX.

The `strcpy_s()` and `strcat_s()` functions can still result in a buffer overflow if the maximum length of the destination buffer is incorrectly specified.

References

[ISO/IEC 99]	ISO/IEC. *ISO/IEC 9899 Second edition 1999-12-01 Programming languages — C*. International Organization for Standardization, 1999.
[ISO/IEC 04]	ISO/IEC. *ISO/IEC WDTR 24731 Specification for Secure C Library Functions*. International Organization for Standardization, 2004.

Pearson Education, Inc. Copyright

Fields

Name	Value
Copyright Holder	Pearson Education

14. The RSIZE_MAX is used to limit the size of objects passed to functions that have parameters of type rsize_t. Extremely large object sizes are frequently a sign that an object's size was calculated incorrectly. For example, negative numbers appear as very large positive numbers when converted to an unsigned type like size_t. Also, some implementations do not support objects as large as the maximum value that can be represented by type size_t. As a result, it is sometimes beneficial to restrict the range of object sizes to detect potential vulnerabilities.

Fields

Name	Value
is-content-area-overview	false
Content Areas	Knowledge/Coding Practices
SDLC Relevance	Implementation
Workflow State	Publishable